オペレーティングシステム SUPER-UX

Operating System SUPER-UX

中西浩一* 岡本 明* 堀 健一*
Koichi Nakanishi Akira Okamoto Kenichi Hori
高橋千恵子* 福島武司** 熊本千秋***
Chieko Takahashi Takeshi Fukushima Chiaki Kumamoto

要 旨

スーパーコンピュータ SX-4 シリーズのオペレーティングシステム SUPER-UX は SX-3, SX-3R シリーズよりの実績ある SUPER-UX をさらに強化し SX-4 向けに最新鋭の機能を搭載したものです。

従来よりの特長である使い勝手の良さ，充実した運用管理，高信頼性を受け継ぎつつ，いっそうの高速・大規模システムへの対応を実現しています。

シングルノードモデルでは最大 32 CPU をサポートしマルチノードシステムにおいては最大16ノードをクラスタ接続することにより最大 512 CPU までサポートします。この広範囲のシステムのスケーラビリティを可能とするために柔軟な資源管理，カーネル・I/O の高い並列処理性，およびマルチノードシステムにおけるシングルシステムイメージを実現するクラスタ制御機能を開発しました。

The SUPER-UX of SX-4 Series provides the most advanced super-computing environment which has been enhanced from the matured SUPER-UX of SX-3/SX-3R Series.

The SUPER-UX can realize higher-performance and larger-scale system retaining user friendliness of UNIX, substantial system administration functions and high reliability of the existing SUPER-UX.

Single system node supports maximum 32 CPUs, while multi node system supports maximum 512 CPUs by clustering up to 16 nodes.

NEC has developed flexible resource control, high level parallelism of kernel and I/O, and clustering control function that realizes a single system image in a multi-node system in order to achieve a wide range of system scalability of the SX-4.

1. はじめに

スーパーコンピュータはハードウエア技術の進歩による計算能力の向上，コストパフォーマンスの向上とともに低価格機の投入により利用分野が拡大し，官公庁中心の利用から民間の各業種へと導入が進みつつあります。

これは科学技術計算分野においてSMP (Symmetric Multi Processer) サーバ領域から分散並列ジョブ実行環境を必要とするハイエンド領域までスーパーコンピュータが必要とされる分野がますます拡がりつつあることを示しています。

したがって単なる高速化技術のみでなく，システム導入，運用およびプログラム開発環境の構築をワークステーション環境と親和性を保ちながら，より容易にかつ柔軟に行えること，また多くのアプリケーションプログラムが動作できることがスーパーコンピュータにとって重要です。

SX-4/SUPER-UX は SX-3/SUPER-UX からの継続性/一貫性を保ちつつ SX-4 のシステム構成に応じた性能のスケーラビリティの実現，運用性向上や新装置サポートのための機能の継続強化，またよりいっそうの標準化/オープン化をめざし API (Application Interface)/ミドルウエア/プラットフォーム標準への準拠の推進を行っています。

本稿では大幅に強化した高並列化処理，CPU/メモリ資源の効率的管理/制御，高速 I/O 制御，高速ネットワーク，NQS (Network Queuing System) 負荷分散処理そしてマルチノードシステムにおけるクラスタ制御などを中心に紹介します。

2. 概 要

SUPER-UX は UNIX† System V をスーパーコンピュータ向けに強化した最新鋭のオペレーティングシステムです。

* 第一コンピュータソフトウェア事業部
1st Computers Software Division
** NEC ソフトウェア東北 第二システム事業部
NEC Software Tohoku, Ltd.
*** NEC ソフトウェア 基本ソフトウェア事業部
NEC Software, Ltd.
† UNIX は X/Open カンパニーリミテッドが独占的にライセンスしている米国ならびに他の国における登録商標です。

以下にSUPER-UXの特長を述べます。

（1） オープンシステム指向

SUPER-UXはAPI/プラットフォーム標準へのタイムリーな準拠を基本として様々な標準機能を積極的に取り入れています。

API，コマンドインタフェースとしてPOSIX，SVID4，SVR4.2MPをサポートし，分散コンピューティング環境としてOSF/DCE，ONC+，分散ファイルシステムとしてDFS，NFS*V3をサポートしていきます。

ネットワーク標準としてはTCP/IPを基本として最新のインターネットプロトコル，たとえばIPngやBGP4を取り入れていく予定です。

（2） 高速性の追求

1ノード当たり最大32プロセッサのサポートに対応し性能のスケーラビリティを得るためにカーネル，入出力処理制御の並列性を大幅に強化（カーネルスレッドによる割り込み処理の並列化，オブジェクトごとの細粒度ロック）しカーネルからライブラリに至るまで効果的な並列動作が可能となりシステム全体の高性能化を実現しています。

またファイルシステムとしてバッファレスデータ転送，ファイル領域の連続割り当てにより入出力性能の大幅な高速化を実現したSFS（Supercomputing File System）や大容量入出力装置に適したSFS/H（Hybrid SFS）を提供します。さらにSFSに対してキャッシュ機能やディスクストライピング機能をサポートした高速入出力サブシステムIAS（Intelligent I/O Accelerator Subsystem）により高速な入出力を実現しています。

さらに高速入出力装置，ネットワーク装置としてHIPPI（High Performance Parallel Interface）インタフェースによるHIPPIスイッチ，ディスクアレイ(RAID)，マスデータプロセッシングシステム(MDPS)，高速画像処理装置(HIPS)およびATMをサポートします。

（3） 多様な運用への対応

システム資源（CPU，メモリ）を効率的に管理するためメモリ/CPUに対する予約型の資源管理(パーティショニング)を導入し，様々な運用形態に応じて柔軟/かつ動的にシステム資源を割り当てることを可能としました。

SX-4では大型，超大型システムのみでなくサーバクラスのエントリシステム(小型，シングルCPU，小容量メモリ，XMUレス)をも提供します。インストレーションを簡易化し短期間での導入を可能としました。

マルチノードシステムにおいては統合コンソールによる全ノードの電源制御/システム立ち上げ/監視，グローバルファイルシステム，全ノードで統一したアカウント/予算管理，NQSのノード間負荷分散によりシングルシステムイメージ（SSI）を実現しています。

* NFSはSun Microsystems, Inc.の商標です。

3. システム制御

3.1 並列処理制御

並列処理は，1つのプロセスを複数のスケジュール単位に分割して，それぞれのスケジュール単位が複数のプロセッサ上で同時に動作することにより実現されます。このスケジュール単位は一般にはスレッドと呼ばれていますが，SUPER-UXではこれをタスクと呼んでいます。タスクはプロセスの資源（アドレス空間やファイル記述子など）を共有し，CPU資源割り当ての単位として独立にスケジュールされます。スーパーコンピュータのように，大規模な演算を行うコンピュータシステムにおいては，単一CPUでは達成できない高速な演算を行うために並列処理は欠かせない機能になっています。

SX-3において並列処理制御はタスクスケジューラというライブラリによって行われていました。このタスクスケジューラはFORTRANにおける並列処理制御を主な目的として設計されていました。そのため，C言語からの利用には一部制限があり，FORTRANプログラムに比べると，並列処理の恩恵を十分に受けることはむずかしいというのが現実でした（図1）。

並列処理は以前はスーパーコンピュータのように大規模な演算を必要とするシステムのための技術であり，特にFORTRANプログラムを中心に使用されていました。しかし，近年では，マルチプロセッサ技術が，ワークステーションやサーバにも当然のように採り入れられるほど身近なものになり，また科学技術計算だけではなく，システムプログラミングでの並列処理も行われるようになってきました。

このような技術の変遷に伴い，並列処理のAPIについても標準化が進められてきました。1995（平成7）年にはスレッドに対するプログラムインタフェースの規格である，IEEE Std. 1003.1c-1995（POSIXスレッド）の標準化が予

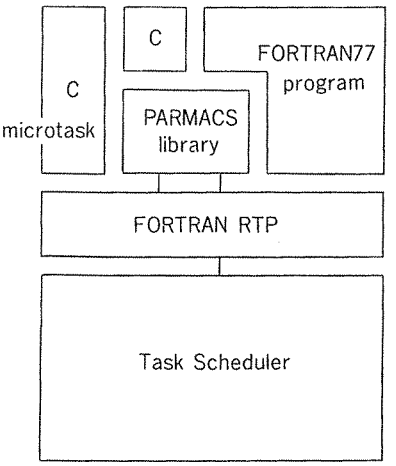

図1 SX-3タスクスケジューラ
Fig. 1 SX-3 task scheduler.

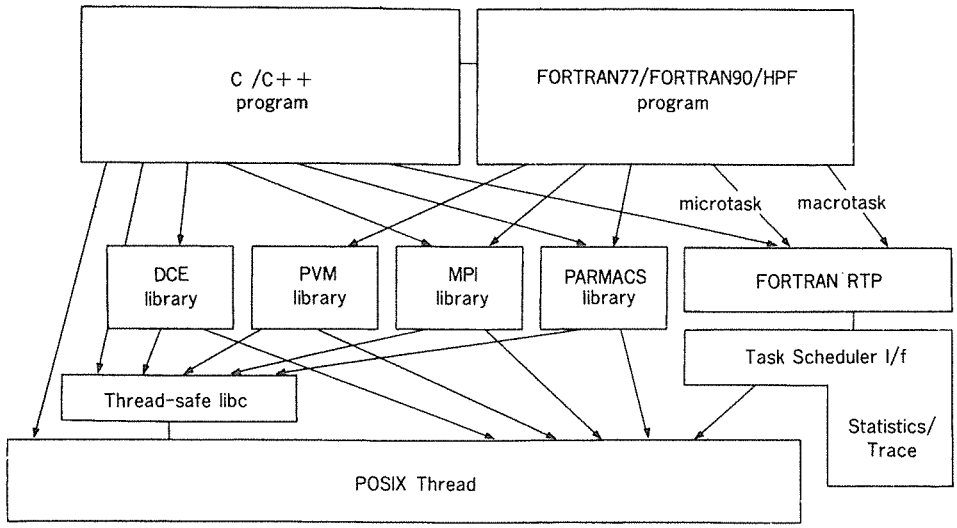

図2　SX-4 並列処理
Fig. 2　SX-4 parallel processing.

定されています。SUPER-UX ではこの規格に準拠した高性能なスレッドライブラリを提供しています。

POSIX スレッドライブラリは単独で利用できるばかりではなく，メッセージ転送ライブラリとして普及しつつある MPI, PVM, PARMACS などや，分散処理環境である，OSF/DCE を実現するための基盤として位置づけられています。また，言語においても，FORTRAN だけではなく，C, C++, FORTRAN90, HPF などを自由に組み合わせて並列処理を可能としています。SUPER-UX のタスクスケジューラも POSIX スレッドライブラリをベースにしたものになっています（図2）。

SX-4 はシングルノード内で最大32台もの CPU を実装することができます。その並列性能を十分に発揮することができるように，POSIX スレッドライブラリには様々な工夫が行われています。

たとえば，SX-3 では，タスクスケジューラ内の資源の保護は単一のロックで行われており，複数の CPU で同時にライブラリ内の処理を行うことができませんでしたが，SX-4 のスレッドライブラリはロックの細分化が行われているので，ロックの競合による待ち合わせを極力抑えることができます。

また，SX-4 では，並列処理のためにアトミック命令が新設されています。スレッドライブラリは，この新しい命令を使用することで，並列処理で重要な同期制御の高速化を実現し，オーバヘッドを劇的に縮小することに成功しています。

3.2　資源制御

一般的な UNIX オペレーティングシステムにおいて CPU やメモリといった資源は一元的に管理されています。この方式において，資源の割り当ては「早いもの勝ち」で行われます。また資源制限を行う場合は，実行単位であるプロセスごとの CPU 時間またはメモリの使用量を制限するといった方式が取られます。

SX-4 では超大規模の演算プログラムが何重にも実行され，かつ対話型のプログラムも同時に実行されます。このような運用を行う場合，従来のようなプロセスごとの資源制限だけではジョブの実行を保障することはできません。また対話型のプログラムの量が数多く実行されると長時間 CPU を使用するジョブがスワッピングされたり，場合によってはメモリ不足によってジョブがアボートする危険もあります。

これに対し SUPER-UX では予約型の資源制限を行う「リソースブロック機能」を提供しています。これは，あらかじめ資源をいくつかに分割し，プロダクションジョブやバッチジョブ，あるいはインタラクティブといった性格の異なるプロセス群に，それぞれ割り当てるといったものです。この機能により，互いの負荷に影響を受けないスケジューリングが可能となり，また高負荷時のジョブアボートを防止することができます。

予約の対象となる資源は以下の3つです。分割された資源をリソースブロックといいます。

① メモリ（ラージページ）*
② メモリ（スモールページ）
③ CPU

これらの資源を分割することにより，メモリにおいてはジョブの実行がメモリ不足で失敗することや，インタラクティブプロセスのメモリ負荷によってバッチジョブがスワップされ実行が妨げられるといったことを防ぐことができます。CPU においては，特定ジョブの CPU 台数を保障し，

* SUPER-UX ではコマンド用にスモールページ（32 K バイト），プロダクションジョブ/一般用にラージページ（4 M バイトまたは 1 M バイト）と2種類のサイズのページを提供しています。

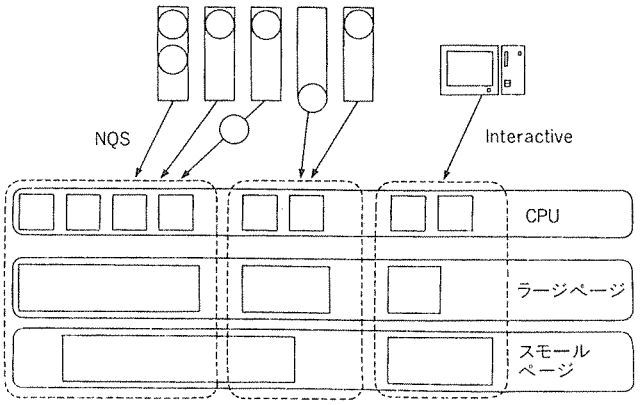

図３　リソースブロック機能の概念図
Fig. 3　Resource block image.

ほかのプロセスの負荷により実行が妨げられるということを防ぐことができます（図３）。

（１）　資源の決定

投入されたジョブやプロセスが，どのリソースブロックを使用するかは，NQS の定義によって決ります。よってユーザ自身が資源の分割を意識する必要はありません。

（２）　柔軟性をもった制限

リソースブロック機能には，「資源の貸し借り」というメカニズムがあります。これは分割した資源であるリソースブロックを絶対的な定義とせずに柔軟性を持たせたものです。

たとえばプロダクションジョブがまったく投入されていないのにバッチジョブが同時にたくさん動いているような場合，プロダクションジョブ用に割り当てた資源をバッチジョブが使用することができます。バッチジョブに資源を貸している状態でプロダクションジョブが投入された場合は，その資源はすみやかにプロダクションジョブに返還されます。

この「資源の貸し借り」のメカニズムにより，あるリソースブロックの負荷が高く別のリソースブロックの負荷が低いといった状態においても，システム全体の資源を有効に使うことが可能となります（図４）。

（３）　サイトに合った運用

リソースブロックの定義は静的なものではなく，動的に変更することが可能です。たとえばプロダクションジョブの流れている日中とプロダクションジョブの流れていない夜間とで設定を変えるなど，よりサイトの実情にあった資源制限を行うことができます。

4.　大規模/高速ファイルシステム

スーパーコンピュータで実行されるアプリケーションプログラムには大容量のデータを扱うものが多くあります。SUPER-UX では，大規模ファイルの高速な入出力を実現しています。

また，SX-3 においては，ローカルファイルシステムとして S5FS, SFS, SFS/H, HXMU の４種が存在していましたが，これを SFS, SFS/H の２種に統合し，ファイルシステムの構築を簡素化しています。以下にこの SFS および SFS/H について詳しく述べます。

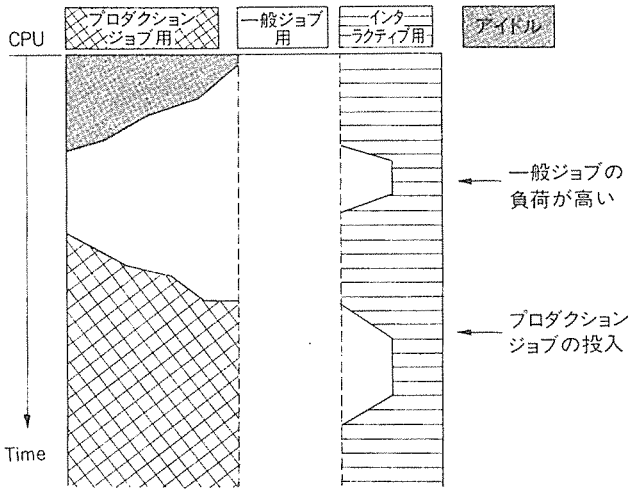

図４　CPU 割り当ての遷移図
Fig. 4　CPU partitioning.

4.1　スーパーコンピュータファイルシステム

（１）　SFS（Supercomputing File System）

SFS はサイズ４K バイトの固定長の領域ブロックで分割管理されています。SFS ではブロックの連続割り当ての最大単位としてクラスタを定義します。つまりクラスタ長（ブロック数）が割り当てる連続ブロックの最大個数となります。

SFS のファイル情報を格納する i ノードを sfinode と呼びます。この形式は，10個の直接アドレッシングのクラスタ指示子，１回の間接アドレッシングのためのクラスタ指示子，２回の間接アドレッシングのためのクラスタ指示子，３回の間接アドレッシングのためのクラスタ指示子がそれぞれ１個ずつあります（図５）。

たとえば，クラスタ長を1,024個（クラスタのサイズとしては４M バイト）とすると次のような巨大なファイルが提供可能となります。

１）直接アドレッシング可能なファイル
　　サイズ：40 M バイト
２）一重間接アドレッシング可能なファイル
　　サイズ：2,048 G バイト
３）二重間接アドレッシング可能なファイル
　　サイズ：２の20乗 T（テラ）バイト
４）三重間接アドレッシング可能なファイル
　　サイズ：２の39乗 T バイト

このような大規模ファイルをサポートするにはファイルが複数のボリュームにまたがって構成される（マルチボリュームファイルと呼ぶ）必要があります。これを可能にするのが仮想ボリューム機能（第４章第２節の仮想ボリュームを参照）です。.

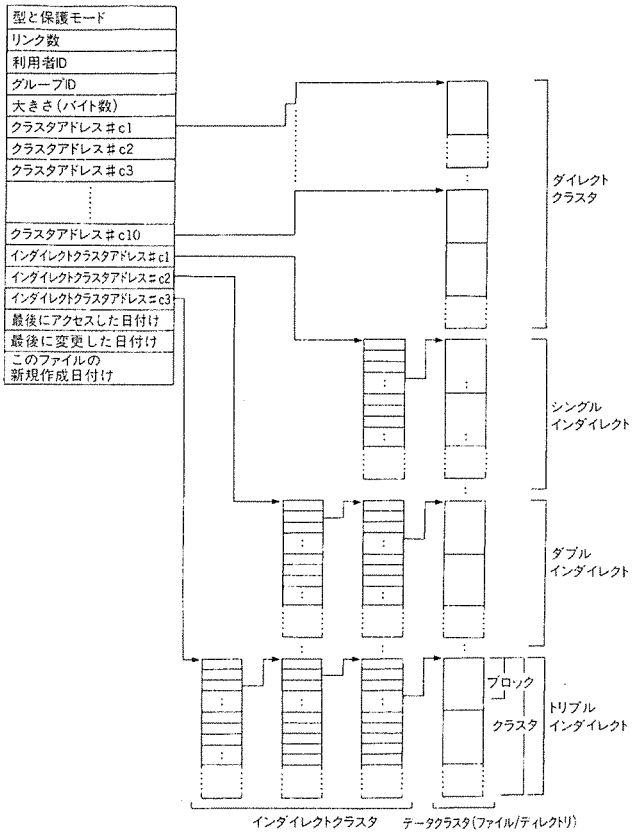

図5　スーパーコンピュータファイルシステム（SFS）
Fig. 5　Supercomputing file system.

SFSの入出力は連続したブロックの集まりであるクラスタのサイズで一度に行うことができます。

たとえば，磁気ディスク装置で複数トラック分のサイズをクラスタとして定義した場合，主記憶装置と磁気ディスク装置の転送において磁気ディスク装置の回転ロスが生じることなく一度に転送します。不連続に存在するブロックを入出力する場合に比べはるかに高速な入出力が可能です。

（2）　SFS/H（Hybrid SFS）

SFS/Hは，ディスク装置やXMUなどの記憶装置のハードウェア性能を最大限に引き出し，高速な入出力を実現するために開発されたファイルシステムです。SFS/Hは，ファイル管理情報やディレクトリデータなどを格納する管理部と，ファイルのデータを格納するデータ部に分かれ，それぞれ別パーティションに格納されます。管理部は，小さなI/Oが多いため，4Kバイトのブロック単位で管理し，データ部は，大容量のファイルデータをなるべく連続した領域に格納するため，複数のブロックをまとめたクラスタ単位で管理します。データ部を格納している装置の特性に合わせてクラスタサイズを設定することにより，装置の性能を引き出すことが可能です。SFS/Hが入出力高速化のために実装している各機能について以下に説明します。

1）連続領域割り当て

SFS/H上のデータ領域は，クラスタ単位で管理され，ファイルへの領域割り当てもクラスタ単位で行います。このためクラスタサイズまでの大きさのデータは，一括して入出力することができます。さらに，複数のクラスタにまたがった領域を割り当てる場合は，できる限り連続したクラスタを割り当てることにより入出力の高速化を図っています。

2）プリアロケート

前記の連続領域割り当ての機能を有効に利用するために，ファイルへの書き込み前にあらかじめ使用するデータ領域を確保する機能（プリアロケート機能）をもっています。

この機能により，複数のクラスタを使用するファイルに対して，前もってすべて連続した領域を割り当てることも可能です。

3）ストライピング

SFS/Hのデータ部は，最大8個のパーティションから構成することができます。これらのパーティションは，ストライピング構成にすることができます。この場合，ファイルのデータは分割されて各パーティションに格納され，それらに対して並列に入出力を行うため，高速な入出力処理ができます。

4.2　仮想ボリューム

仮想ボリューム（Virtual Volume）は，単一長ブロック（4Kバイト）で構成される論理的な記憶装置です。ユーザプログラムからは，あたかも以下のような記憶装置があるようにみえます。

・大容量記憶装置
・ストライピング型記憶装置
・キャッシュ付き記憶装置

しかし実際には，これらはIASが提供する次の機能によって実現されるものです。

・マルチボリューム機能
・ストライピング機能
・キャッシュ制御機能

また，記憶装置のスペースを効率良く利用するためのリアロケーション機能も提供しています。以下に各機能について説明します。

1）マルチボリューム機能

標準UNIXでは複数の磁気ディスク装置にまたがってパーティションを作成することができません。したがって作成できるファイルのサイズは物理的に1つの磁気ディスク装置の容量によって制限されることになります。この問題を解決し，複数の磁気ディスク装置からなるパーティションの作成を可能とするのがマルチボリューム機能です。これにより，複数の磁気ディスク装置や拡張記憶装置を組み合わせて大容量のファイルシステムを作成することができます（図6）。

2）ストライピング機能

ストライピング機能は，ファイルデータを複数のディスク装置に分散して格納し，それらのディスク装置へ並行し

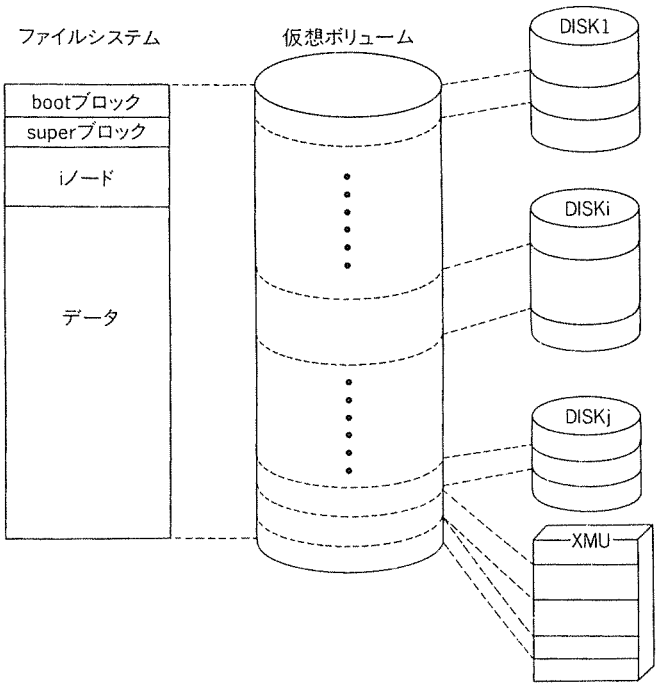

図6　マルチボリューム機能
Fig. 6　Multi volume function.

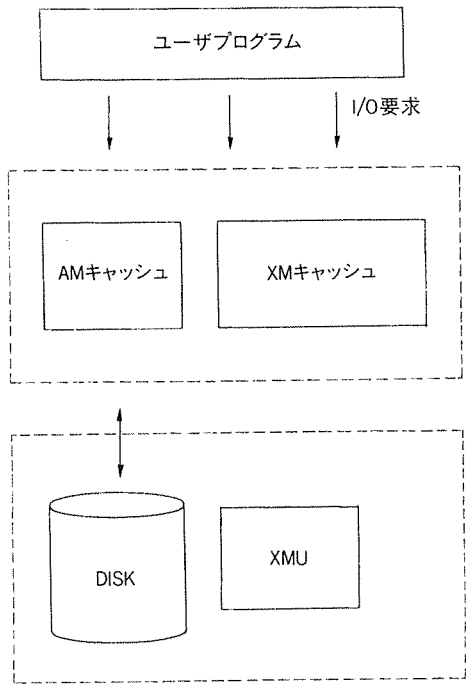

図8　キャッシュ機能の構成
Fig. 8　Configuration of cache control facility.

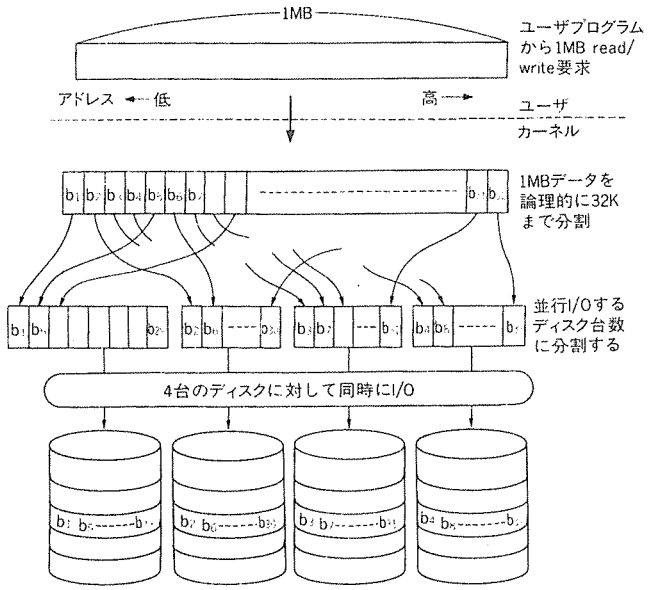

図7　ストライピング機能
Fig. 7　Striping function.

て入出力を行うことにより，入出力性能を向上させる機能です。大きなデータを一度に入出力する場合に効果を発揮します（図7）。

3）キャッシュ制御機能

ある種のディスク装置は，制御装置内にディスクキャッシュと呼ばれる高速の半導体メモリを持ち，アクセス頻度の高いデータだけをディスクキャッシュに格納して入出力の実行性能を高めています。このディスクキャッシュ方式を磁気ディスクからなる仮想ボリュームに適用したものが仮想ボリュームキャッシュ機能です。

仮想ボリュームキャッシュは主記憶装置（MMU），拡張記憶装置（XMU）から構成されます。これらをそれぞれAMキャッシュ，XMキャッシュと呼びます（図8）。仮想ボリュームキャッシュ領域はSTU（Staging Unit）と呼ばれる128Kバイトの単位で管理されています。キャッシュ上のSTUはアクセス頻度によってプライオリティ付けされ，キャッシュ領域を有効に利用するように制御されています。

4）リアロケーション機能

SFSでは仮想ボリュームとしての物理的な連続領域であるクラスタを領域の割り当て単位としていますので，ファイル領域が物理的に連続となり，入出力が高速化されます。しかし逆に領域の割り当て単位が1クラスタに満たない場合には，スペース効率が低くなります。そこで，クラスタを仮想化し，実際の割り当てはその単位をブロック→STU→クラスタと移動させることによってスペースの効率化を図るリアロケーション機能を実装しています。データの移動は，割り当て単位を越えて書き込み要求が発生した時に行われます（図9）。

4.3　大規模記憶領域管理

大規模記憶領域管理として，ファイルアーカイビングシステムSX-BackStoreを提供しています。

本システムは，ファイルをほかの大容量ストレージへマイグレーション（データをコピーし，もとのデータを解放）することで，普段利用されるファイルシステムの空き領域を増やします。そして，ユーザがそのファイルをアクセス

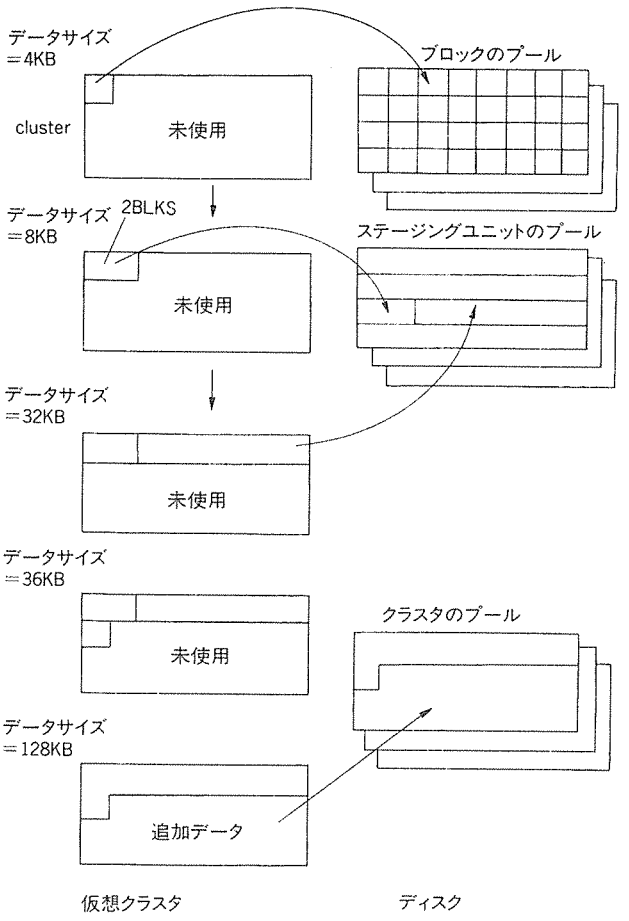

図 9　リアロケーション機能
Fig. 9　Virtual volume reallocation.

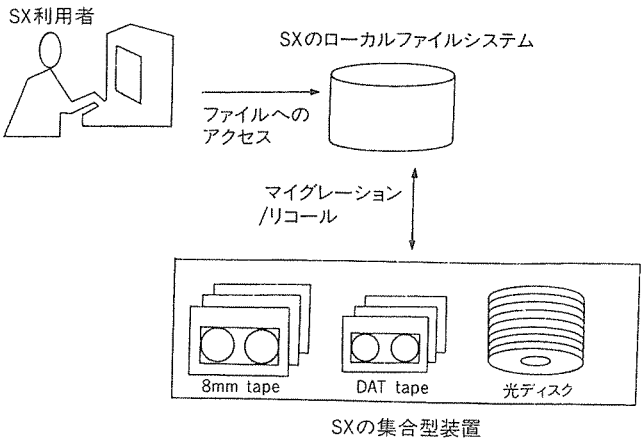

図 10　SX-BackStore（サーバ型）
Fig. 10　SX-BackStore (server).

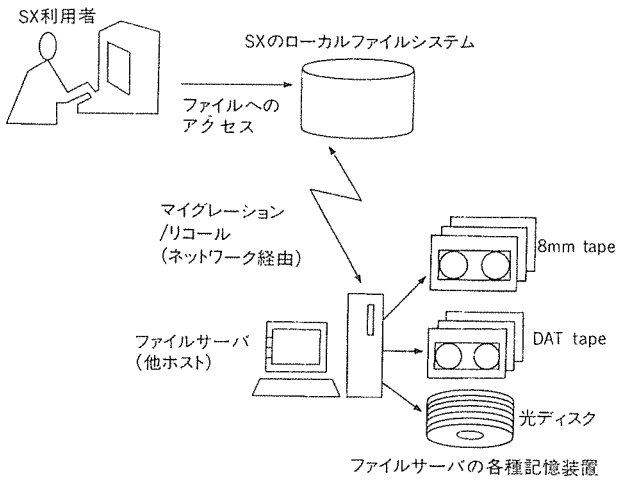

図 11　SX-BackStore（クライアント型）
Fig. 11　SX-BackStore (client).

したときは，データを自動的にリコール（もとのファイルシステムへのデータの復帰）します。これらの処理はシステムによって自動的に行うことが可能なため，ユーザは本システムを意識する必要がありません。また，従来使用していたファイルシステムの再構築は不要ですし，本システム導入によるアプリケーションの変更も必要ありません。

本システムにより，通常利用しているファイルシステムを仮想的にＴバイト以上の容量を持つファイルシステムとして利用することができます。これによって，ファイルシステム容量不足によるジョブのエラーなどを極力防ぐことが可能となります。

また，頻繁にアクセスされるファイルはそのまま高速にアクセスできますし，アクセスの頻度が低いファイルはマイグレーションすることで効率的なファイルシステムの運用が可能です。

マイグレーションされたデータを格納するストレージとしては，SX システムに直結されるマスデータプロセッシングシステム（MDPS）や集合型装置（現在計画中）と他ホストで稼働するファイルサーバシステムがネットワーク経由によって利用できます（図 10，図 11）。

5.　高速入出力制御

5.1　XMU 制御

SX-4 の XMU（拡張記憶装置）は，SX-3 の XMU と比較して性能向上をはじめとする種々の機能拡張や改善が行われています。具体的には，記憶素子に 16 M ビット DRAM を採用し，ノード当たり最大容量 32 G バイト，データ転送速度は最大 16 G バイト/秒の高性能を実現しています。また，４台の XCU（XMU-Channel control unit）に接続された XMU とのデータ転送がそれぞれ並列に動作できるので，システムスループットが向上しています（図 12）。

そのハードウェア性能を十分に生かすために，SUPER-UX では以下の機能を新規に実装しています。

（１）　非同期入出力制御

SX-3 では，XMU と主記憶とのデータ転送はすべて同期型，すなわちデータ転送処理中は CPU を独占していました。これに対し SX-4 では非同期入出力がサポートされ，データ

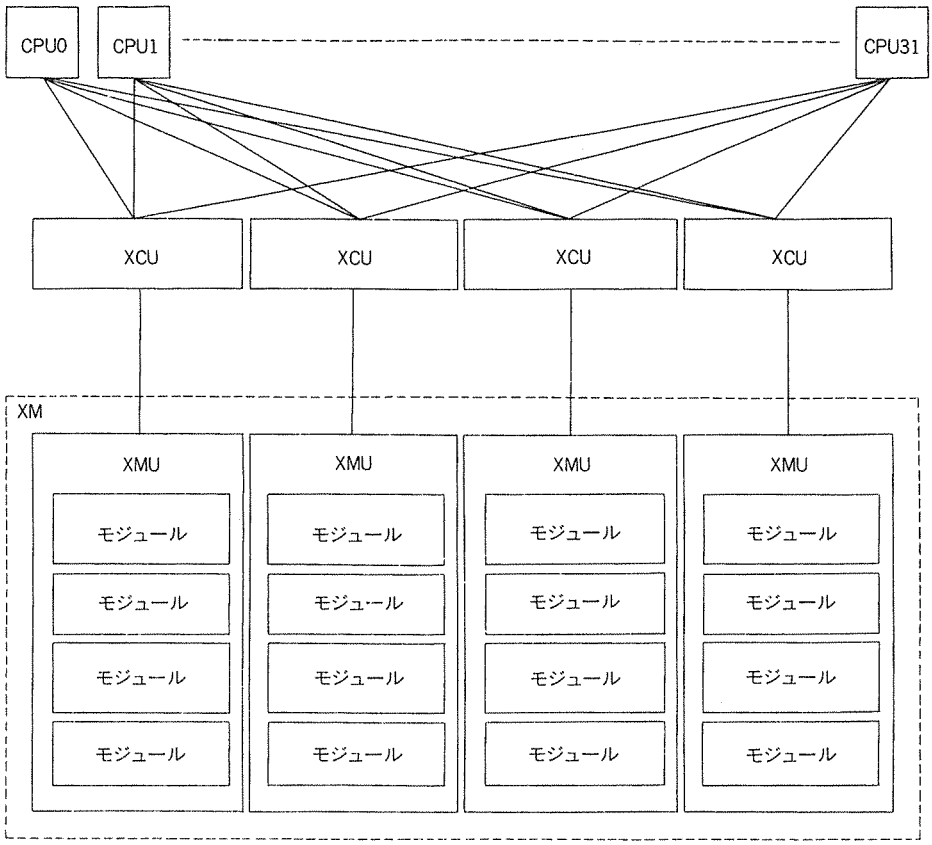

図 12　XMU 構成図
Fig. 12　Configuration of XMU.

転送中に CPU を他プロセスに割り振ることが可能になるため，特に XMU との大量のデータ転送を含むプログラムの並列性が向上します。さらに，個々の入出力要求に対して非同期型で処理するかどうかはオペレーティングシステムが自動的に判断して最適な手段を選択してくれるため，利用者はまったく意識する必要はありません。

この制御は，主にオペレーティングシステム内の XMU ドライバが行っています。XMU ドライバは，発行された入出力要求のデータサイズがある「しきい値」より大きい場合には，そのデータ転送の完了を待つ間にほかのプロセスに CPU を与えた方がいいと判断し，非同期入出力命令を発行します。逆に，データサイズが「しきい値」より小さく，転送完了までの時間よりもプロセスを切り替えるためのオーバヘッドの方が大きいと判断される場合には，従来どおりの同期入出力を行います。この「しきい値」は，利用者により各 XMU 装置ごとに指定することが可能です。

（2）　非特権入出力制御

XMU に対するデータ転送命令を，システムコールを経由してオペレーティングシステムから発行する従来の方式のほかに，ユーザプログラムから直接発行するためのインタフェースを提供しています。この機能により，システムコールを発行することで発生するプログラムオーバヘッドを大幅に削減し，XMU のもつハードウェア性能をよりいっそう効果的に引き出すことができます（図 13）。

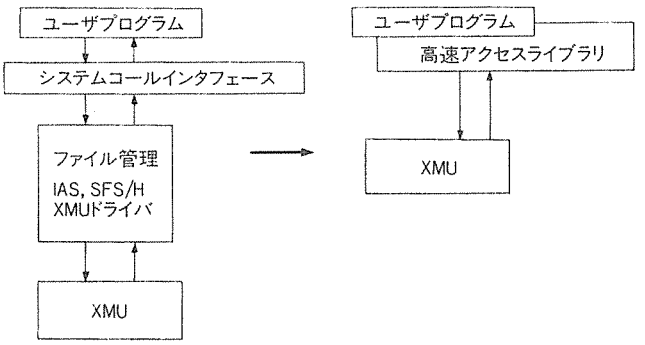

図 13　XMU 非特権アクセス
Fig. 13　XMU nonprivileged access.

本インタフェースは，FORTRAN の RTP にリンクするライブラリとして提供され，XMU のパーティションによって構成された SFS/H ファイルシステムに対する入出力をいっそう高速に行えるようにします。

この機能を用いて入出力を行う場合，ユーザプログラムから直接 XMU に対してデータ転送命令が発行されるため，オペレーティングシステムによる領域の排他制御が行われません。そこで，あるプログラムがアクセスする XMU 上の領域を前もってハードウェア側に通知しておくことによ

り，プログラムのミスなどによる不正領域へのアクセスを防止しています。

（3） XMU 配列機能

FORTRAN プログラムの配列データを，主記憶上だけでなく XMU 上にも格納することが可能です。したがって，主記憶の容量を超える大規模なデータを扱うプログラムの実行が可能です。

さらに，前述（2）の非特権入出力機能を用いれば，XMU 上に格納された配列データに対する，より高速なデータ処理を行うことができます。

5.2 HIPPI 制御

SUPER-UX では，各種記憶装置や他ホストとの高速入出力を実現するために，ANSI 準拠の超高速インタフェースである HIPPI を採用しています。HIPPI のデータ転送能力は，800 Mbps と大変高速であり，接続される周辺入出力装置のハードウェア性能を十分に生かすことができます。さらに，ノード当たり最大64本（32ペア）の HIPPI チャネルを接続可能なので，多彩な周辺入出力装置を同時に接続して使用することができます。

また，遠隔接続用の光ファイバによる HIPPI エクステンダや HIPPI スイッチに接続することにより，大規模な高速ネットワークを構築することができます。これについては後述します。

SUPER-UX では以下の周辺入出力装置に対する HIPPI 入出力をサポートしています（**図 14**）。

（1） 高速ディスクアレイ装置（RAID）

75 M バイト/秒のデータ転送能力をもつ RAID タイプの磁気ディスク装置です。記憶容量は，サブシステム当たり 128 G バイトと大容量です。

（2） マスデータプロセッシングシステム（MDPS）

高速・超大容量の光磁気ディスク装置です。最大データ転送速度は 10 M バイト/秒，サブシステム当たり約 1.6 T バイトの大容量を備えています。

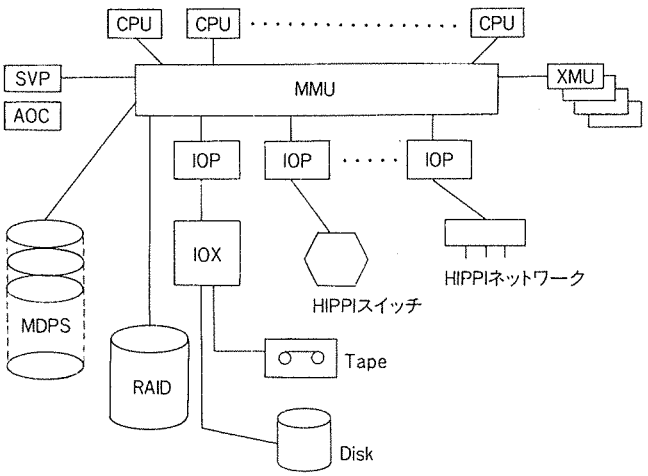

図 14 HIPPI 入出力装置
Fig. 14 HIPPI I/O devices.

（3） 入出力制御装置（IOX）に接続された各種記憶装置

以下の記憶装置をサポートしています。

① ディスクアレイ装置
② 4 mm DAT 装置
③ 集合型 DAT 装置
④ 8 mm カートリッジテープ装置
⑤ ¼インチカートリッジ磁気テープ装置
⑥ ½インチカートリッジ磁気テープ装置

（4） 高速画像処理装置（HIPS）

画像データを高速転送することにより，計算結果をリアルタイムに視覚化して確認することができる高速画像処理装置です。

SX-4 の HIPPI 入出力制御では，高速ディスクアレイ装置の持つコマンドスタック機能（複数の入出力要求パケットを同時に受信できる機能）を有効に利用するために，パケットの先出し制御を行っています。この制御によって，複数の入出力要求をまとめて装置に送り，オペレーティングシステム内の処理を先に進めることができ，CPU 資源を有効に使うとともに，一括転送によるデータ転送速度の向上が期待できます。

また，ある入出力装置に対して異なるデータ転送パスを複数設定しておくことにより，チャネル障害など何らかの理由で特定のパスが使用できなくなった場合に自動的に代替パスへの切り替え処理を行います。これによって，障害時に入出力装置が使用不能になることを極力回避しています（**図 15**）。

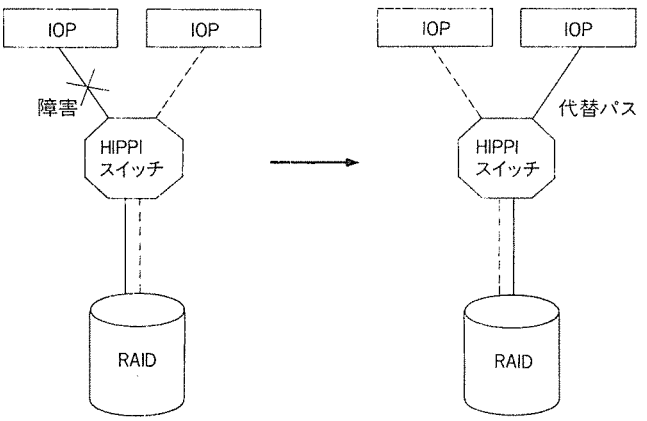

図 15 代替パス機能
Fig. 15 Alternative path function.

6. バッチ処理

6.1 NQS (Network Queuing System)

NQS は，アメリカ航空宇宙局（NASA）の航空力学数値シミュレーション（NAS）システムの計画の一環として，ネットワークに接続された多種多量の UNIX マシンを統一して使用できるようにするために，Sterling Software 社によって開発されたネットワーク対応のリクエストキューイ

ングシステムです。

SX-3では，このシステムを基に各種の機能拡張を行っており，SX-4においてもSX-3において拡張した機能はそのまますべて継承しており，クラスタシステムにおいては，シングルシステムイメージ(SSI)のユーザビューを提供するため，新たに以下の機能をNQSに組み込んでいます。

・負荷分散機能

・NQSジョブトラッキング機能

以下の第2節，第3節では，新規機能である上記2つの機能について説明します。

6.2 負荷分散機能

(1) 機　能

NQSの一般的な処理系では，一括処理するコマンドの組合せであるバッチリクエストを基本処理単位として扱います。

このリクエストをNQSに投入することで，NQSの機能を利用することが可能となります。NQSは受け付けたリクエストをキューと呼ばれるものにいったん溜めておき（キューイング），リクエストやキューの属性にしたがって順次リクエストを選択（スケジューリング）して実行していきます。先のバッチリクエストを専門に処理するキューをバッチキュー，リクエストをほかのキューに転送するキューをパイプキューと呼びます。パイプキューがリクエストを転送するキューは，ローカルホスト上にあるものはもちろん，ネットワークを介したリモートホスト上にあるものでも可能です。逆に，ネットワークを介したリモートホストのキューへリクエストを投入する場合は，パイプキューを通じて行わなければなりません。

クラスタシステムにおいては，このローカルホストとリモートホストの関係がクラスタシステム内のノード間の関係となります。

通常パイプキューのリクエスト転送先には，複数のキューを設定することができます。パイプキューにリクエストが投入されると，パイプキューにあらかじめ設定されているそれらの転送先キューを順にチェックし，リクエストの属性と転送先キューの属性や状態に応じて転送先キューが選ばれ，転送が行われることになります。

しかしこの場合，リクエストの特性がほぼ一定のものであれば，転送先はその設定された順序に依存するようになり，同じ転送先にばかり転送が行われる可能性が高くなります。

この結果として，リクエストの実行結果を得るまでのターンアラウンドタイムが悪くなり，ひいてはマシン間の負荷バランスの効率が悪くなるなど，クラスタシステムとしては好ましくない状況となってしまいます。

これに対応するため，パイプキューに設定されている転送先の中から，リクエストの実行結果を得るまでのターンアラウンドタイムが最も早くなる転送先をNQSが自動的に選択してリクエストを転送するのが負荷分散機能です（図16）。

NQSにおけるキューの負荷判定要件としては，

・キューにおける実行リクエスト数（キューの多重度）

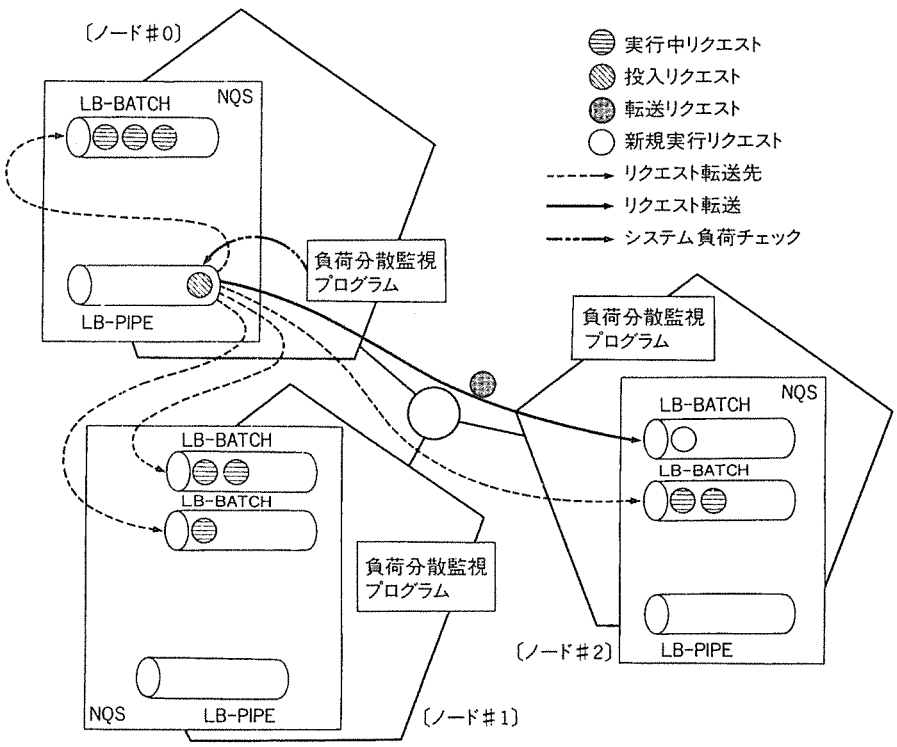

図16 負荷分散機能

Fig. 16 Load balancing facility.

・キューに割り当てられた使用可能資源総量とリクエストの資源使用要求量
・リクエスト保有ユーザ/グループのクラスタシステム内での予算
があり，システムにおける負荷判定要件としては，
・キューに対応付けられた資源分割グループにおけるメモリおよびCPUの使用率（絶対残量）
・システムのロードアベレージ
があります。

これらの負荷判定用件をキューごとの属性として，運用サイトの方針に応じて複数の判定式を定義できるようにし，利用者は，リクエスト投入時にどの判定式を用いるかを指定するようにします。キュー属性に何も定義されていない場合は，システム提供の方針を適用します。

この負荷分散機能の導入により，NQSシステムの負荷バランスが自動的に効率良く保たれ，NQSシステム管理者の負荷監視作業の負担が軽減できるなどのメリットも期待されます。

（2） 負荷分散専用キューの導入と負荷分散環境におけるキュー構成

クラスタシステムにおいては，通常のパイプキューやバッチキューとは別に，負荷分散専用のパイプキュー（負荷分散パイプキュー：以降LB-PIPEと表記），および負荷分散専用のバッチキュー（負荷分散バッチキュー：以降LB-BATCHと表記）を準備しています。LB-PIPEの転送先にはLB-BATCHのみが設定可能となっています。

これはクラスタシステムとしての負荷分散を有効に機能させるために，負荷分散の有効範囲を，ひとつのクラスタ内に限定するためです。

負荷分散という機能を考えた時，LB-PIPEに設定される複数のLB-BATCHの割り当て可能な資源量といった属性は一致していなければ意味がありません。

これについては，クラスタの管理者がキュー構成を設計し，NQSをインストールする際のツールおよびチェック機構で矛盾が起こりにくくするようになっています。

また，同じLB-BATCHの属性のグループに対し，複数のLB-PIPEを準備し，それぞれのLB-PIPEのキューのプライオリティを適切に設定することで，キューのクラス分けが可能です。ユーザアクセス制限も利用すれば，管理者の緊急ジョブ専用キューなども構築することもできます。

リクエストのプライオリティは従来のNQSと同様，同一のキュープライオリティを持つキュー内のリクエストの選択順序に影響します。

6.3 NQSジョブトラッキング機能

負荷分散機能の導入により，リクエストの転送がNQSによって自動的に行われてしまうと，リクエスト投入者は，自分の投入したリクエストが今どのノードのどのLB-BATCHに転送されたのか，また，そのリクエストは現在どのようなステータスにあるのかといったことを把握することが困難になります。

そのようなリクエストのステータスを把握するため，リクエスト転送をNQSが自動的に追尾するジョブトラッキング機能を提供しています。これにより，リクエスト投入者は転送先を順に検索するなどの手間をかけなくても，リクエストのIDを元に，状態表示，削除や停止などの制御をリクエスト投入元で行うことが可能となります。

ユーザによるリクエストの表示は，表示要求を発行したノード上にあるすべてのユーザのリクエスト，および，そのノードから投入され，ほかのノードに転送されたすべてのユーザのリクエストをトラッキングファイルから検索して表示します。

また，新規オプションにより，リクエスト実行ノード，リクエスト投入ノードの枠を越えて，ユーザリクエスト情報を参照することもできます。

7. 高速ネットワーク

スーパーコンピュータ上で実行される超大型計算では，一般に大量の入出力データが発生します。入力データを外部からシステムに投入したり，結果を高速に外部へ出力するためには，高速ネットワークが不可欠です。特に，科学技術計算の結果をグラフィックとして出力するサイエンティフィック・ビジュアリゼーション(Scientific Visualization)の概念は非常に重要であると考えられていますが，フルカラーのアニメーションを出力できる帯域幅をもつ超高速ネットワークは，これを実現するための強力な手段となります。また，計算センターの中央マシンとして，遠方の不特定のユーザに高度なサービスを提供する必要がありますが，この時，機種にかかわらず自由に相互接続が可能であること，ネットワークトポロジーの変化に容易に追随できることが重要事項となります。

SX-4はマルチベンダ環境の中にも異和感なく溶け込み，高速大容量通信を実現して超高速コンピュータサーバとしての役割を果たすことが期待されています。

7.1 開発方針

SX-4のネットワーク機能は，以上のような要求に応えるために次のような方針で開発しました。

（1） 高速大容量通信の実現
① HIPPIネットワーク
② ATMネットワーク
③ UltraNet*
④ FDDI
⑤ Ethernet**

（2） 標準プロトコル/標準API環境への対応
① TCP/IPプロトコルファミリ
② ソケットインタフェース

---

* UltraNetは米国Computer Network Technologies社の登録商標です。
** Ethernetは XEROX社の商標です。

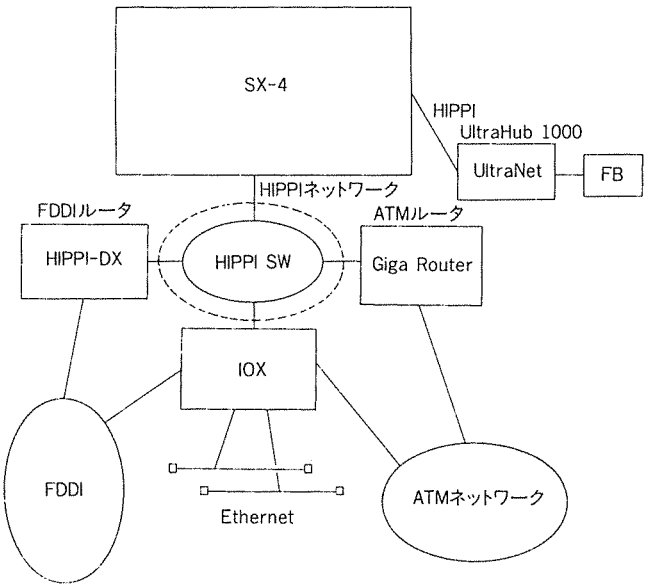

図 17　SUPER-UX ネットワーク環境
Fig. 17　SUPER-UX network environment.

③　TLI（Transport Level Interface）
④　OSI プロトコル

（3）　運用管理の標準化・自動化の推進

①　NetAdmin による統合的ネットワーク運用管理
②　snmpd, gated など，最新のネットワーク管理技術のサポート
③　インターネット標準の積極的取り込み

7.2　各種ネットワークのサポート

以下のネットワークをサポートしています（図 17）。

（1）　HIPPI ネットワーク（LAN）

SX-4 では，ANSI の標準および RFC1374 で規定されている，HIPPI-FP（Framing Protocol），HIPPI-LE（Link Encapsulation）と，IEEE 802.2 で規定されている，LLC（Logical Link Control）とに準拠したパケット形式で HIPPI 上で IP データグラムを送受信する機能をサポートしており，HIPPI スイッチや HIPPI 上の IP ホストなどから構成される IP ネットワークを容易に構築することができます。HIPPI ネットワークは HIPPI Extender により距離的に拡張することもできます。HIPPI は 100 M バイト/秒の転送速度をサポートしており，ネットワーク系の周辺機器やほかの IP ホストとの間で，高速大容量転送が可能です。

（2）　ATM ネットワーク（LAN/WAN）

SX-4 を ATM ネットワークに接続する方法には次の 2 つがあります。

①　IOX の ATM-NIC による接続
②　HIPPI-ATM ルータ（たとえば，NetStar 社の GigaRouter など）による接続

これらにより，RFC1483 で推奨されている，AAL-5（ATM Adaptation Layer）と LLC に準拠した形式での IP データグラムの送受信が可能となり，容易に ATM 上の IP ネットワークに参加することができます。物理層としては，OC-3 C SONET（155 Mbps）がサポートされており，ATM ネットワーク上のほかのホストとの間で，高速大容量転送が可能です。

（3）UltraNet（LAN）

UltraNet は，Computer Network Technologies 社の超高速 LAN で最大 1 Gbps の速度をもっています。SX-4 は Ultra1000 という大型の Hub に HIPPI チャネルで接続されます。これにより，UltraNet に接続されたグラフィックワークステーションやほかのスーパーコンピュータと画像などの大量データを瞬時にやりとりすることができます。

特に Hub に直結されるフレームバッファを用いれば，フルカラーのアニメーションを表示することも可能となり，サイエンテフィック・ビジュアリゼーションの実現に大いに威力を発揮します。

（4）　FDDI

FDDI は 100 Mbps のトークンリング型 LAN で，マルチベンダ環境でのバックボーン LAN としてすでに主流となっています。SX-4 と FDDI とは次の 2 つの方法で接続できます。

①　IOX の UTP-FDDI ボードによる接続
②　HIPPI-FDDI ルータ（たとえば，NSC 社の HIPPI-DX など）による接続

SX-4 では，FDDI/IP プロトコルをサポートしており，FDDI 上の他システムと高速に大量のデータを送受信することができます。

（5）　Ethernet

パーソナルコンピュータを始めとしてほとんどすべての機器でサポートされていますので，最も手軽にネットワークを構築することができます。SX-4 では，TCP/IP プロトコルと OSI プロトコルをサポートしています。

7.3　プロトコル/API

以下をサポートしています（図 18）。

（1）　インターネットプロトコル

TCP/IP を含むインターネットプロトコルは，SX-4 のメインプロトコルとして，フルサポートされています。

特に SX-4 では，スーパーコンピュータにふさわしい機能と性能のために，以下のことが特徴的です。

1）IP の論理限界である 64 K バイトのデータグラムを TCP，UDP，IP で扱える。

2）最大 64 K バイトのバッファを使用可能である。

3）強制的に大きな MSS（Maximum Segment Size）で TCP の通信を行うオプションが利用可能。

4）64 K バイトより大きな Window Size を用いた通信を可能とする TCP のオプションが利用可能。

また，SX-4 では，本格的インターネットワーク環境の中でも十分に力を発揮できるように，最新のプロトコル規格（RFC：Request For Comments）に基づいたフィードバックが常に行われています。将来的には，次世代 IP である IPv6

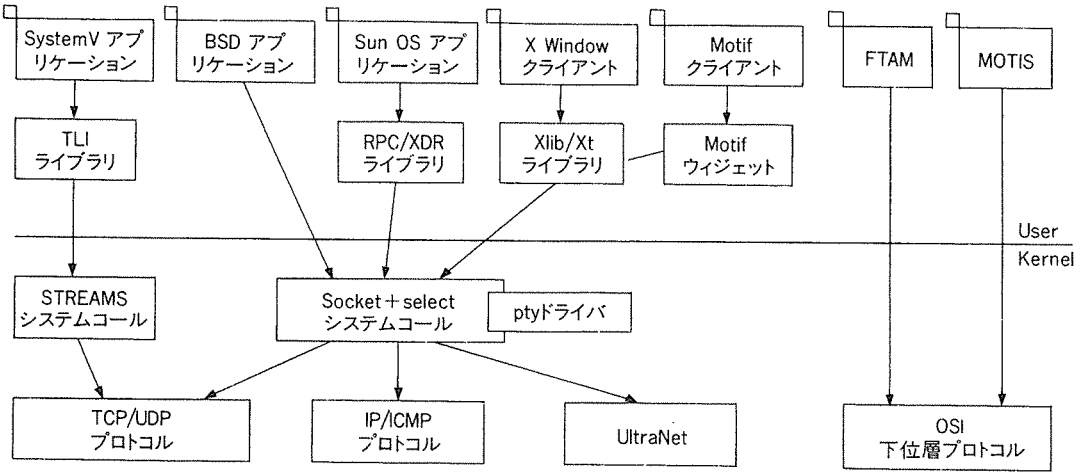

図 18　SUPER-UX API サポート
Fig. 18　SUPER-UX API support.

への迅速な対応，および，OSPF2/BGP4 などの最新のルーティングプロトコルの実装などが計画されています。

（2）　API

以下がサポートされています。

1）ソケット

BSD のソケットシステムコールを実装しています。

2）TLI（Transport Level Interface）

System V 標準機能として，TLI ライブラリをサポートしており，TCP/UDP プロトコルを利用可能です。

3）RPC/XDR

Sun Microsystems 社で開発された NFS（Network File System）や NIS の基盤を成す RPC/XDR の機能をライブラリとして提供しています。

4）Xlib/Xt

X Window* システムがサポートされており，X クライアントの移植/作成が容易に行えます。

5）OSF/Motif**

PSF/Motif のウィジェットや UIL（User Interface Language）を備えており，Motif クライアントの移植/作成が容易に行えます。

（3）　OSI プロトコル

SUPER-UX がサポートする第2のプロトコルは国際標準である OSI です。SUPER-UX としては主に LAN 系の国際標準プロフィルをサポートしています。下位層のプロトコルスタックとしては，TP4，CLNS，ES-IS，LLC type1，CSMA/CD をサポートしています。

上位層は FTAM，MOTIS をサポートしています。

7.4　分散処理アプリケーション

下記がサポートされています。

① telnet/rlogin
② ftp/rcp/rdist
③ NFS
④ NIS
⑤ BIND（named/resplver）
⑥ X Window システム
⑦ NetAdmin
⑧ rsh/on
⑨ gated/routed
⑩ NQS/RQS
⑪ mail（SMTP）
⑫ SNMP（Simple Network Management Protocol）
⑬ lpd

8.　クラスタシステム制御

果てしなく増大する超高速演算ニーズに応えるために複数のノードを接続したマルチノードシステムにおいても，クラスタ制御によりユーザ側からは一つのシステムとして利用できる機能（シングルシステムイメージ（SSI））を実現し，使いやすく快適な利用環境を提供します。

クラスタシステムの全体構成図を図 19 に示します。

（1）　システム操作

統合コンソール機能によって，オペレータは，システム制御 IOX に接続した端末から，システム全体を対象とした全ノード一括および各ノードを対象に以下の操作が行えます。

1）電源制御

本体系装置および周辺系装置に対する電源投入，システム停止（shutdown）に連動した電源切断が可能です。また，特定ノードおよび特定ノードの特定装置の電源制御も可能です。

2）システムの起動・停止

システム電源が投入された状態でのシステム起動はもちろん，電源投入からシステムの起動までの一括操作が可能です。また，システム全体の一括停止指示が可能です。

** X Window システムは，X Consortium, Inc. の商標です。
*** Motif は，Open Software Foundation, Inc. の商標です。

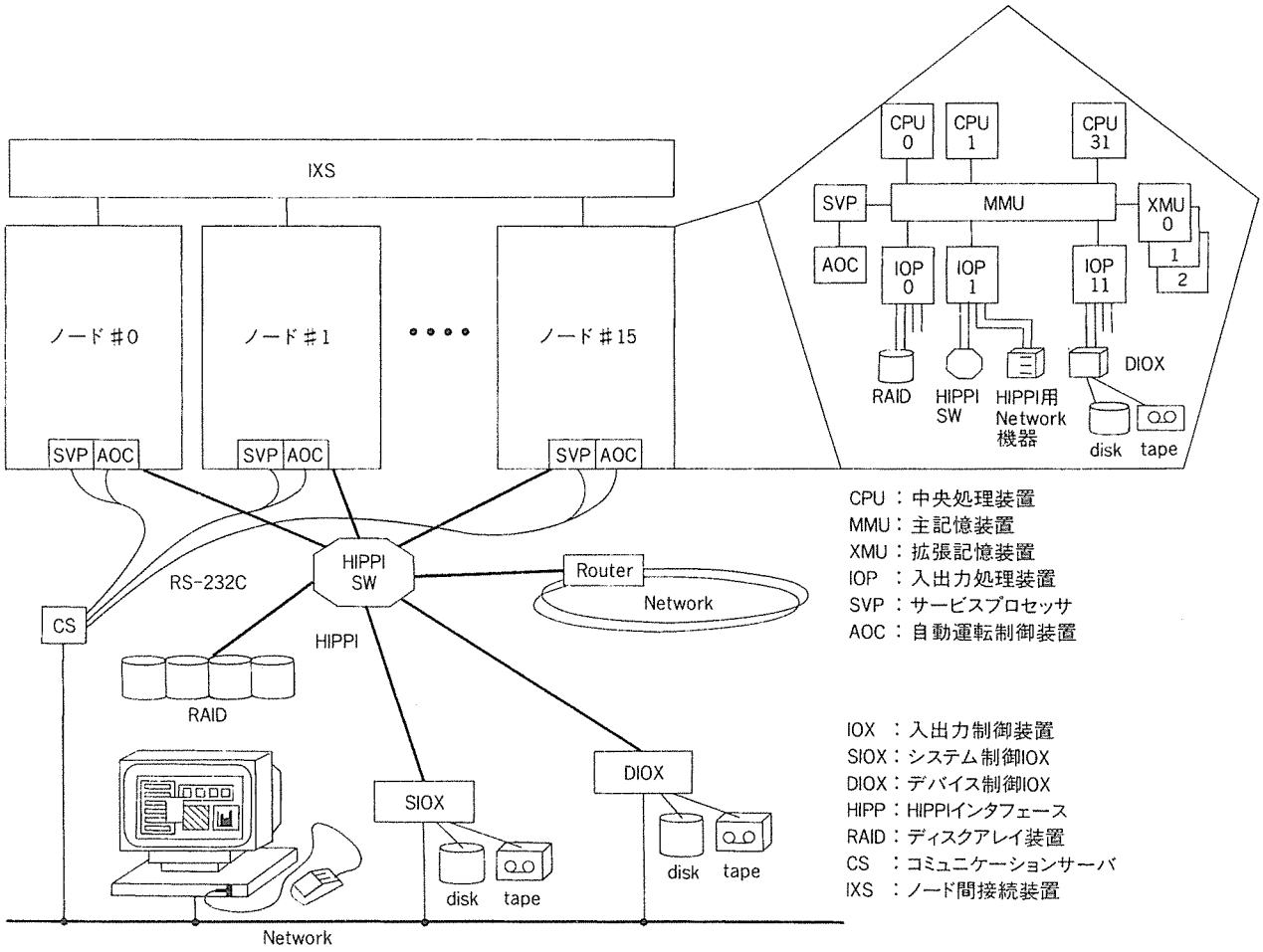

図19　クラスタシステム構成図（IXS接続）

Fig. 19　Configuration of cluster system (IXS connection type).

3）システム監視

システム運用中は全ノードの状態を監視し，障害などが発生した場合，統合コンソール上で警告します。また，各ノードで出力された運用レベルのメッセージやエラーメッセージを集中管理します。メッセージは全ノード一括表示のほか，ノード指定またはメッセージレベルなどを指定して編集表示することができ，システム管理者はシステム全体の運用状況を効率よく監視することができます。

4）コマンド実行

オペレータは運用中の全ノードに対して同一のコマンドのリモート実行が可能です。また，指定ノードに対するコマンド実行も可能です。これらのコマンド実行は対象のノードにログインすることなく実行できます。

5）自動運転

あらかじめ設定された運用カレンダにしたがって，システム全体の電源投入，システムの自動起動およびシステム全体の停止(shutdown)，自動電源切断を行うことができます。また，ある特定プロセスの監視，およびプロセス異常時の自動再起動処理により自動運転を行うことができます。これらの運用カレンダの設定やプロセス監視の設定はシステム一括およびノードごとに異なる設定ができます。

（2）　システム管理

1）時刻管理

クラスタシステムにおいては，各ノード上で採取された各種ログ情報を集約し，時系列に表示するなどシステム内の時刻を一致させる必要があります。本システムでは，NTP (Network Time Protocol) を使用して，システム制御IOXは標準時刻に一致させ，各ノードはシステム制御IOXまたは他ノードに一致させることにより，システム内の時刻同期を実現しています。

2）セキュリティ

各ノードのセキュリティレベルを統一し，ノードに依存しないセキュリティ環境を提供する必要があります。各ノードのセキュリティ情報を一致させるため，システム制御IOX上でセキュリティ情報を一括管理し，同一情報を各ノードに配布します。また，各ノード単位に採取された監査情報を時系列にマージして編集出力することができます。

（3）　利用者管理

1）ユーザ/グループ管理

ユーザはシステム内の任意のノードを指定してログイン

する必要がありますが，どのノードにも同一ユーザ名/グループ名でログイン可能です。また，ユーザ登録および削除はシステム制御 IOX で一括して管理しますので，ノードごとに登録する必要はありません。ログインパスワードなど一般ユーザが各ノード上で変更可能なデータについては，あるノード上での変更が全ノードに反映されます。

2）ログイン管理

ノード管理者はノード単位に特定ユーザのログインを制限することができます。また，全ノードでログイン制御データを同一に設定することでクラスタ全体で同一のログイン制限が可能です。制限できる項目は，以下のとおりです。

① ログインを拒否できるユーザ名
② 資源制限（CPU 使用時間，メモリサイズ，データセグメントサイズ，スタックセグメントサイズ）
③ 同一ユーザが同時にログインできる人数
④ アカウント/予算管理

アカウント情報（プロセスアカウント，セッションアカウント，ジョブアカウント，ファイル転送アカウントなど）はノード単位に採取され，集計ファイルが作成されます。このノードごとの集計ファイルをマージしてクラスタ全体の集計を行い，クラスタ全体での日報/月報の作成を行います。また，クラスタ全体でのユーザ（またはグループ，アカウントコード）ごとのシステム料金を計算して，あらかじめ設定した各々の予算超過を監視することができます。システム全体の予算設定はシステム制御 IOX 上で行います。また，システム全体の予算を超過した場合，全ノードにおいて，ログイン時およびジョブ投入時にシステムの使用制限が行われます。

（4） ファイル管理

1）グローバルファイルシステム（図 20）

クラスタシステムでは投入されたジョブは各ノードの負荷状況に応じて最適なノードに転送される負荷分散機能を提供します。このため，ユーザのディレクトリ/ファイルはすべてのノードから透過にアクセスできるように統一された名前空間を提供する必要がありますが，本システムでは NFS のクロスマウント機能を利用して実現します。このようにして構築されたファイルシステムをグローバルファイルシステムと呼びます。このグローバルシステムの構築や維持管理をサポートするために，以下の機能を提供します。

① 自動マウント機能

各ノードの立ち上げ時にグローバルファイルシステムのマウントを自動的に行う機能。

② 代替マウント機能

あるノードがダウンしたとき，当該ノードがローカルファイルシステムとしてマウントしていたグローバルファイルシステムを代替ノードに自動的にマウントする機能。

（5） システム構成管理

1）クラスタ構成制御

クラスタに属するノードはあらかじめクラスタ構成情報として定義されます。クラスタ制御機能はこのクラスタに属するノードとそのノードの状態を意識して制御を行っています。たとえば，クラスタ内のあるノードが停止状態にある場合，そのノードをジョブの負荷分散の対象外とするような処理を行います。クラスタ内の一部のノードを切り離し，そのノード上で異なるシステムの運用や保守を行うことができます。また，運用中のクラスタに影響を与えることなく，切り離されたノードを再度クラスタに組み込むことができます。

クラスタ分割運転の概念図を図 21 に示します。

2）ノード間共有装置排他制御

複数のノードから物理的にアクセス可能な装置で，論理的にも複数のノードからアクセスされる可能性のある装置

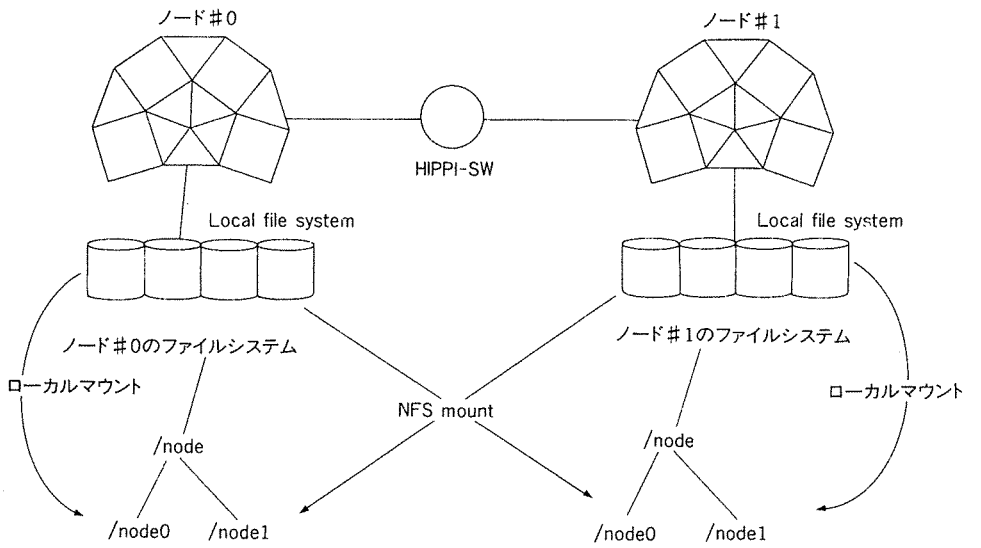

図 20 グローバルファイルシステム
Fig. 20 Global file system.

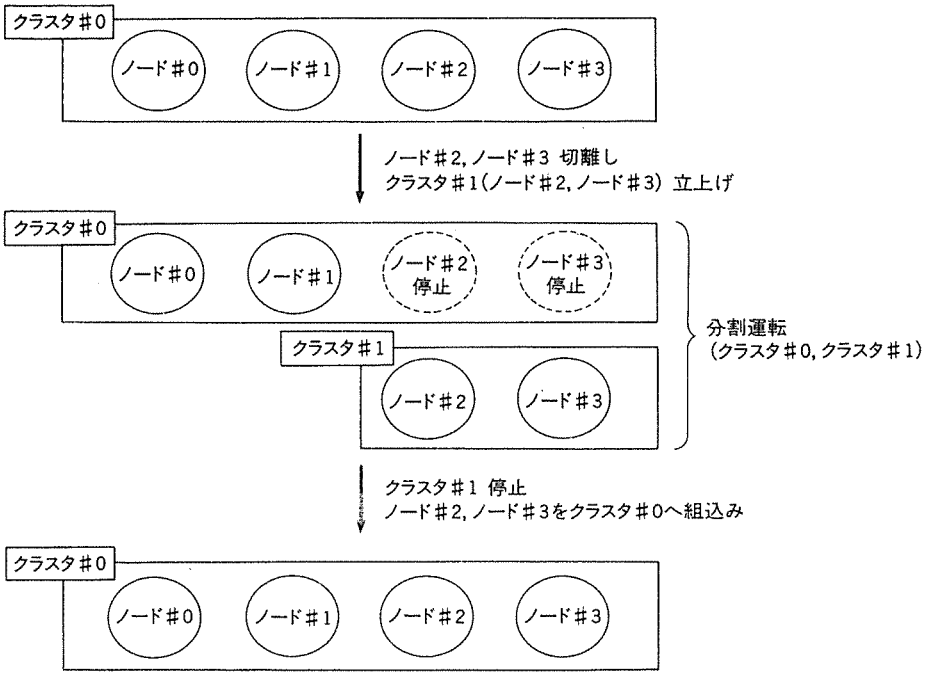

図 21　分割運転
Fig. 21　Divided operation.

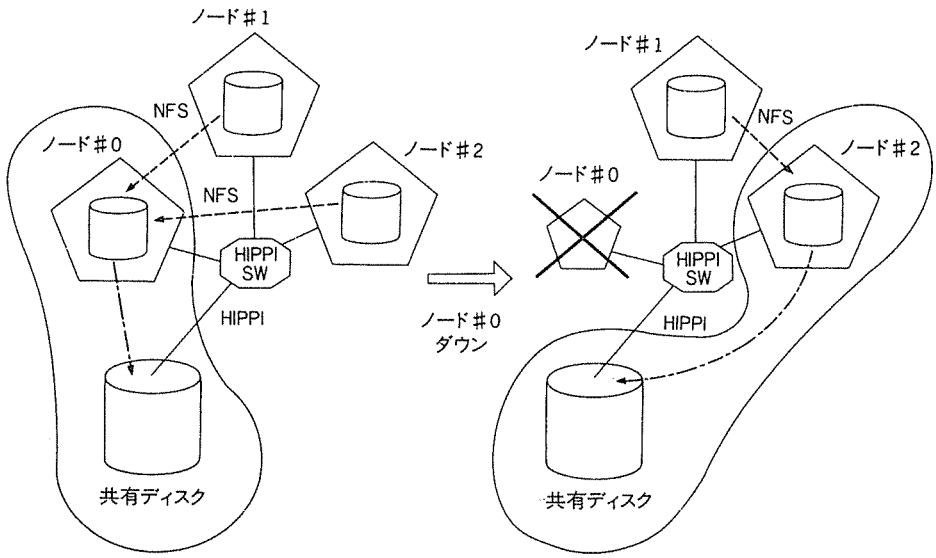

図 22　共有ディスクのノード引継処理
Fig. 22　Inheritance of shared disks.

をノード間共有装置と呼びます。ただし，ある一定期間は単一のノードに論理的に割り付けられ，そのノードが占有使用できます。言い換えればほかのノードからはアクセスできないように排他制御されます。ノード間共有装置のノードへの割り付けはシステム制御 IOX 上の制御コマンドによって行います。たとえば，ノード間共有装置の管理主体であったノードがダウンした場合，別のノードに割り付け，以降，その装置上のファイルの管理を引き継ぐことでファイルを継続してアクセスすることが可能です（図 22）。

（6）　障害管理

1）ノード障害処理

単一のノードが障害となった場合，障害となったノードは自動的にクラスタから切り離され，残りのノードで運用を継続する，いわゆる縮退運転となります。ノードが障害となった事象は統合コンソール上にメッセージ表示され，ノード状態は障害状態と表示されます。なお，ノード障害復旧後は，ノードを再起動し，クラスタに再組み込みすることができます。

2）IOX 障害処理

IOX がダウンしてもクラスタ運用は継続します。ただし

統合コンソールや課金管理などシステム制御IOX上に実装されている機能は一時的に縮退します。IOXが復旧した場合にはシステムを停止させることなくシステムに組み込むことができます。また，ハードウェア固定障害などによるIOXダウンした場合のIOX機能の早期回復のためにIOXの二重化機能を提供します。待機系のIOXは通常は一般のUNIXサーバとして使用され，運用系の障害時には新運用系となります。また，障害復旧したIOXは新待機系となります。なお，切替えには，自動切替とコマンドによるマニュアル切り替えの2種類を提供します。

（7） システム生成・保守

1）システムインストール

全ノード一括の自動インストール機能をサポートします。本機能はノードの起動順序を記述したノード起動ファイルとノード内でのインストール手順を記述したノードインストールスクリプトの組合せによって実現されます。

2）バージョンアップ

全ノード一括のバージョンアップおよび各ノード個別のバージョンアップが可能です。個別バージョンアップではクラスタ内の該当ノードをクラスタから切離し(shutdown)バージョンアップ後，もとのノードに再組み込みすることを全ノードにおいて実施します。これにより，クラスタシステムとしては無停止でバージョンアップが可能となります。

9. 高信頼性機能

SUPER-UXは，大規模化したスーパーコンピュータの信頼性の向上を図るため，各種の障害に対する迅速な復旧を行うとともに，その影響範囲を局所化するなどの豊富な高稼働性機能を提供しています。

（1） 再試行，回復処理

システムの障害をできるだけ早期に検出し，その回復を図ります。

本体系装置や周辺系装置の障害検出と再試行などの回復処理を行います。また，入出力処理装置の冗長化構成システムにおいては他系による回復処理を行い，システム運転を継続することができます。

（2） 構成制御

障害による影響を局所にとどめ，システム運転の継続を図ります。

システムを構成する装置の構成状態を管理し，障害回復ができない装置の自動切り離しによる影響範囲の局所化および障害回復時の自動再組み込みによる再構成処理を行い円滑なシステム運用を可能とします。

（3） 障害情報記録

システムを構成する装置の障害は，エラーログとしてファイルに蓄積されます。これらの情報は障害箇所を発見するため，あるいは装置や入出力媒体ごとのエラー発生状況の把握による予防保守のための情報として使用され，障害箇所の早期発見や障害が致命的状態になる前に保守を行うことを可能にしています。

（4） システム自動再立ち上げ

ハードウェアの致命的な障害などによりシステムの継続運転ができない場合，一度システムを停止した後，障害装置を自動的に切り離してシステムの自動再立ち上げを行います。

（5） メッセージ自動応答

システムあるいはユーザプログラムから発行されたメッセージを契機に自動リカバリを図るメッセージ自動応答機能を提供しています。本機能によりシステム障害を未然に防止したり，障害を最小限に抑えることができます。

（6） サブシステム監視

ある特定のプロセスの監視を行い，そのプロセスが異常終了した場合はリカバリ処理の後，再起動する機能を提供しています。

10. むすび

スーパーコンピュータSX-4は1 GFLOPSから1 TFLOPSまでの1,000倍の性能スケーラビリティをもち，低価格の小型エントリーシステムから本格的な分散共有メモリ並列処理システムまで多様なニーズに対応できる画期的な製品です。

オペレーティングシステムSUPER-UXも限りない性能要求と，より柔軟で容易なシステム運用への要求に応えるべく大幅な機能強化を行いました。

今後も低価格化と高性能化によりスーパーコンピュータの適用分野はますます拡がるものと思われます。

市場動向，技術動向を見極めつつ顧客のニーズに一歩先んじながらSUPER-UXの開発に努力する所存です。